Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 370 (2007) 279–292
Theoretical Computer Science
www.elsevier.com/locate/tcs

Feasibility and complexity of broadcasting with random transmission failures☆

Andrzej Pelc[a], David Peleg[b,*]

[a] *Département d'informatique, Université du Québec en Outaouais, Gatineau, Canada*
[b] *Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel*

Received 1 August 2006; received in revised form 28 October 2006; accepted 31 October 2006

Communicated by G. Ausiello

Abstract

Fault-tolerant broadcasting in the message passing and radio models is considered under a probabilistic failure model. At each step, the transmitter of each node may fail with fixed constant probability $p < 1$, and failures are independent. Both node-omission and malicious transmission failures are studied. Our goal is to establish conditions on feasibility and to estimate the (synchronous) time complexity of almost-safe broadcasting (i.e., broadcasting which is correct with probability at least $1 - 1/n$ for n-node graphs and for sufficiently large n) under these scenarios. If only node-omission failures are assumed, almost-safe broadcasting is feasible for any $p < 1$, in both communication models. For malicious transmission failures, almost-safe broadcasting in the message passing model is feasible iff $p < 1/2$, and in the radio model it is feasible iff $p < (1 - p)^{\Delta+1}$, where Δ is the maximum degree of the network. For the time complexity of almost-safe broadcasting, a number of upper and lower bounds are given in the various models.

Keywords: Broadcasting; Malicious transmission failures; Node-omission failures; Message passing; Radio networks; Random failures

1. Introduction

1.1. Background and models

As computer networks grow in size and become more vulnerable to component failures, it becomes increasingly important to design communication algorithms that work correctly in spite of some limited number of failures and without knowing their locations.

A faulty scenario in network communication is mainly determined by three aspects: the duration of failures, their type and their distribution (cf. the survey [25]). In particular, failures can be either permanent (a component remains faulty during the entire execution of the communication process) or transient (the failure concerns a transmission in a given step only). In this paper we restrict attention to transient failures. Specifically, we concentrate on *transmission* failures, which may be either *node-omission* failures or *malicious transmission* failures. When a node-omission failure

☆ An extended abstract of this work has appeared in the proceedings of the 24th ACM Symposium on Principles of Distributed Computing, July 2005.

* Corresponding author.
E-mail addresses: pelc@uqo.ca (A. Pelc), david.peleg@weizmann.ac.il (D. Peleg).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2006.10.031

occurs at a given step, the faulty node does not send any messages during that step. Such failures are relatively benign; although some information may be lost, at least the information that is received can be trusted. Malicious transmission failures, on the other hand, represent a worst-case fault scenario: such a failure can cause the transmission component of a faulty node to behave arbitrarily, by either stopping, or altering transmitted messages in a way most detrimental to the communication process. It can also transmit in steps in which the algorithm requires it to remain silent). The model allows adaptive adversarial behavior, namely, one depending on the execution's history. Granted, malicious transmission failures that exhibit all these kinds of damaging behavior rarely occur in practice (although they may be caused, for instance, by a hostile agent whose aim is to destroy the communication process). Nevertheless, the concept of malicious failures plays an important role in the study of communication algorithms, as it represents a worst-case assumption: communication algorithms that work correctly in the presence of malicious failures can be used safely under any fault scenario.

The two most commonly studied models of failure distribution are the *bounded* model and the *probabilistic* model. In the bounded model, an upper bound k is imposed on the number of faulty components and their worst-case location is assumed. In the probabilistic model, failures are assumed to occur randomly and independently of each other, with some specified probability distribution.

In this paper we study one of the most important communication primitives, which is *broadcasting*. This primitive requires one node of the network, called the *source*, to disseminate a message to all other nodes of the network. The manner in which the message is transmitted through the network depends upon the particular communication model employed. In this paper we consider two common synchronous communication models, known in the literature as the *message passing model* and the *radio model*. In the message passing model a node can send, in each time step, arbitrary and possibly different messages to all of its neighbors simultaneously. Any node can receive all messages sent to it by its neighbors in that time step.

The radio model is more restrictive. A node can transmit at most one message in any given time step, and that message gets delivered to *all* neighbors. A node receives a message in a given step if and only if it does not transmit in this step itself and exactly one of its neighbors transmits in this step. Otherwise a collision occurs, which the receiving node cannot distinguish from silence (we assume no collision detection capability).

We adopt the following fault scenario. In every step, the transmissions of each node fail with constant probability $0 < p < 1$. Transmission failures of different nodes are independent, and so are transmission failures of the same node in different steps. We study both node-omission and malicious transmission failures. It should be stressed that we distinguish between two separate components of the node, namely, its main control processor and its transmission component. (Such a separation often exists in modern network architectures, in which switching and other communication-related tasks are carried out by a fast dedicated "switch" or "router", cf. [10].) The type of malicious behavior assumed here is limited only to the transmission component of the node and does not affect, e.g., the memory or control components of the nodes. In particular, a failed node does not tamper with its own memory registers and internal state. Hence if the node does not fail in subsequent steps, it regains normal behavior. While this is admittedly a deviation from the standard notion of malicious failures, it may be natural for studying transmission failures.

Let us remark that while the fault model is assumed to be probabilistic, the algorithms presented are deterministic. This means that the environment can behave randomly, but given a particular behavior of the environment, the execution of the algorithm is fully determined. (Nevertheless, our impossibility results hold for randomized algorithms as well.)

In a failure-prone environment, the task of broadcasting is completed successfully if, despite the failures present in the network, every node receives a copy of the original source message intact. Note that, since we adopt a probabilistic scenario, the best we can achieve is broadcasting with high probability. A broadcasting algorithm working under a given fault and communication scenario in an n-node graph G is said to be *almost-safe* if it manages to complete broadcasting succesfully with probability at least $1 - 1/n$, for sufficiently large n. Note that an almost-safe broadcasting algorithm might still *err*, i.e., fail to deliver the source message to some nodes, albeit with negligible probability of at most $1/n$.

1.2. Our results

The goal of this paper is to establish conditions on feasibility and to estimate the (synchronous) time complexity of almost-safe broadcasting under four scenarios, obtained from combining the message passing or

radio communication models with node-omission or malicious transmission failures. We consider faults affecting only the transmission components of nodes. For the resulting four scenarios we show the following. Assuming node-omission failures, almost-safe broadcasting is feasible for any $p < 1$, in both communication models. Assuming malicious transmission failures, almost-safe broadcasting is feasible iff $p < 1/2$ for the message passing model and it is feasible iff $p < (1-p)^{\Delta+1}$, where Δ is the maximum degree of the network, for the radio model. For the time complexity of almost-safe broadcasting we give the following upper and lower bounds.

Consider an n-node graph G with a given source s, and denote by D the radius of G w.r.t. s (namely, the largest distance from s to any node in G). Then for the message passing model we show that assuming node-omission failures, the optimal almost-safe broadcasting time is $\Theta(D + \log n)$. Assuming malicious transmission failures, almost-safe broadcasting is possible in time $O(D + \log^{\alpha} n)$, for any constant $\alpha > 1$.

For the radio model we cannot expect time $\Theta(D + \log n)$, as even fault-free broadcasting schemes (constructed by polynomial algorithms) have time larger than $\Theta(D + \log n)$ for some graphs [15]. Hence a natural benchmark is *opt*: the value of fault-free broadcasting time for a given graph. We show that almost-safe broadcasting in time $O(opt + \log n)$ is impossible for some graphs, even with node-omission failures, and we give an almost-safe broadcasting algorithm with time $O(opt \cdot \log n)$ for any graph, for both types of failures.

1.3. Related work

For a general survey of fault-tolerant broadcasting and gossiping see, e.g., [25]. More specifically, broadcasting with malicious transmission failures in the bounded fault model was studied, e.g., in [5,14], and in the probabilistic model in [6,8,9]. However, in the latter papers, failures were assumed permanent and were distributed randomly once for the entire communication process. On the other hand, broadcasting with randomly distributed malicious transmission failures on a line were studied in [23]. Omission transmission failures in the probabilistic model were investigated, e.g., in [13].

Broadcasting in the radio model without failures was studied, e.g., in [3,7,12,21], while fault-tolerant broadcasting in this model was the subject of [20,22]. Fault-tolerant broadcasting under the t-locally bounded model, where at most t permanent malicious failures are allowed in the neighborhood of every node, was introduced and studied in [20]. In [22] failures were assumed of permanent omission type.

Protocols for communication over a single unreliable link (known as *data link protocols*) were studied from a distributed algorithmic viewpoint in [24,16,1]. In [16] it was shown that in a worst-case (adversarial) setting, no deterministic protocol can tolerate host crashes, even when the channel guarantees FIFO and no duplication. Subsequently, a randomized solution was developed in [24].

The problem of fast message transfer over a path of unreliable processors was studied in [19]. The model studied is semi-synchronous, i.e., where there are known lower and upper bounds on message delay, and the algorithm relies on detecting and isolating failures. In the presence of malicious failures, this requires the additional assumption of some cryptographic means (e.g., a secure signature scheme). The model is not probabilistic, hence the paper makes no attempt to quantify the probability of failure. Rather, the focus is on optimizing the communication complexity of algorithms that guarantee optimal time.

Reliable communication in dynamically changing networks was studied in [4] and subsequently in [17,2]. The model studied in [4] is semi-synchronous, with a lower and upper bound on the message delay on each link. It is assumed that nodes are failure-free but the topology may change arbitrarily, in the sense that links may fail and recover infinitely many times, so long as the network is *eventually connected*, i.e., no edge-cut persists forever. It is also assumed that when a node sends a message on an incident link and the link fails, the node will be informed of this fact within bounded time. In this setting, the paper develops efficient algorithms for broadcasting or routing a sequence of messages reliably, namely, in order and without omission. Subsequent papers dealing with the end-to-end message transmission problem in dynamic networks of this type [17,2] discard the semi-synchrony assumption and attempt to bound the size of counters used in the solution, while still maintaining low message complexity.

All of the above studies follow a worst-case rather than probabilistic failure model and hence address complexity issues differently than in this paper.

2. Feasibility constraints

In order to establish the feasibility of broadcasting, we consider naive broadcasting algorithms and ignore efficiency considerations. In these algorithms, broadcasting is performed along a spanning tree of the network rooted at the source s, with each node receiving the message from its parent in the tree.

We present two algorithms, one for node-omission and the other for malicious failures. For simplicity and uniformity of presentation, each algorithm is formulated so that it works in both the message passing and the radio models. In particular, to avoid collisions in the radio model, the algorithm activates only one transmitter in each step.

2.1. Node-omission failures

In a preprocessing stage, construct and fix a spanning tree T of the network rooted at the source s. (This can be done centrally using a standard algorithm [11].) Let $v_1, \ldots, v_n$ be an enumeration of all nodes of the tree, ordered by nondecreasing distance from s in T. Thus the enumeration respects the levels of T. Let c be a constant depending on p, to be determined later, and let $m = \lceil c \log n \rceil$. It is assumed that n and p, hence also c and m, are known to the nodes.

Whenever our algorithms specify that a node v should transmit a message M in step t, this is interpreted as stated in the radio model, and in the message passing model it is taken to mean that v should send M to each of its children in the tree T.

Algorithm Simple-Omission

For $i = 1$ **to** n **do**
Phase i**:** **For** m steps:
– v_i transmits the source message M_s (or 0 if it has not received M_s).
– All other nodes remain silent.

As described, the algorithm requires each node to identify the "window" of time steps during which it should transmit. To facilitate the identification, the nodes should have a common notion of time (namely, a global clock). In the absence of simultaneous wake-up, this can be achieved by the sender initiating a step counter to 0 and attaching it to its transmitted messages, increasing the counter at each round, thus allowing every node receiving a message for the first time to synchronize with the others.

Moreover, the identification is based on the assumption that each node v_i knows its index i. In the message passing model, this assumption can easily be discarded by slightly modifying the algorithm. Essentially, a node will start its window of transmissions upon receiving the message for the first time. In the radio model, if the nodes have no distinct labels (i.e., the network is anonymous) then broadcasting is impossible for some networks, like the 4-cycle, due to symmetry. Assuming distinct labels from some range $[0, K-1]$, with every node knowing its label, it is possible to enforce that at most one node transmits in every round and each node gets sufficiently many opportunities to transmit. This can be done, for instance, by relying on a global clock established as explained above, and requiring a node with label i to transmit only in time steps $\ell K + i$ for integer $\ell \geq 0$ (or alternatively – in case K is unknown to the nodes – in time steps p_i^k for integer $k \geq 1$, where p_i is the i'th prime).

We have the following.

Theorem 2.1. *Assume node-omission transmission failures that occur with probability $p < 1$. Then almost-safe broadcasting is feasible in both the message passing and the radio models.*

Proof. We show that Algorithm Simple-Omission performs almost-safe broadcasting for a suitable choice of the constant c. Since in every step only one node transmits, there are no collisions and we give the same argument for the message passing and the radio models. Let c be such that $p^{\lceil c \log n \rceil} < 1/n^2$. Let E_i be the event that node v_i does not fail in at least one of the m steps where it should transmit, and let $E = \bigcap_{i=1}^{n} E_i$. We have $P(\bar{E}_i) = p^m < 1/n^2$ and hence $P(E) > 1 - 1/n$.

Assume that E holds. Then it is easily shown by induction on i that when v_i's turn to transmit arrives on the ith iteration, it has already received the source message M_s, hence it can transmit it. It follows that if E holds then all nodes receive M_s upon completion of Algorithm Simple-Omission. Hence this algorithm is almost-safe. ■

2.2. Malicious transmission failures

2.2.1. The algorithm

The naive algorithm presented next for establishing the feasibility of broadcasting with malicious transmission failures is a version of Algorithm Simple-Omission in which nodes perform additionally a vote on messages received from the parent in the tree and relay the majority result of this vote (or the default value 0 if there is no majority). Also the constant c guaranteeing almost-safe broadcasting is different from the previous case. The tree T and the enumeration $v_1, \ldots, v_n$ of the nodes is as before. Again, $m = \lceil c \log n \rceil$, and it is assumed that n, p, c and m are known to the nodes.

Algorithm Simple-Malicious

The source v_1 transmits the source message M_s for m steps;

For $i = 2$ **to** n **do**
Phase i**:**
- v_i computes $M_i :=$ the majority message among the messages received by v_i from its parent;
- v_i transmits M_i for m consecutive steps.
- All other nodes remain silent.

Whereas almost-safe broadcasting is achievable in the node-omission failure model for any failure probability $p < 1$, the malicious transmission failure model exhibits a threshold of p values for which almost-safe broadcasting can be achieved. Algorithm Simple-Malicious applies to both the message passing and the radio models. However, its analysis is different, and so is the p threshold for almost-safe broadcasting. We therefore present the analysis separately.

2.2.2. Analysis for the message passing model

Theorem 2.2. *Assume malicious transmission failures that occur with probability $p < 1/2$. Then almost-safe broadcasting is feasible in the message passing model by a deterministic algorithm.*

Proof. Using standard arguments based on Chernoff's bound [18], it can be shown that if $p < 1/2$ then for a suitable constant c, every node computes the correct message M_i with probability at least $1 - 1/n^2$. Hence Algorithm Simple-Malicious succeeds with probability at least $1 - 1/n$, thus it is almost-safe. ■

The algorithm requires each node to identify two time windows: first, the "listening window" during which it should expect to receive the message from its parent in the broadcasting process, and subsequently, the "transmission window" during which it should transmit to its children in the broadcasting process. This can be easily achieved assuming each node v_i knows its index i and all the nodes wake up simultaneously at round 0.

The two assumptions can again be discarded in the message passing model by modifying the algorithm and its analysis. First, observe that in this model there is no particular reason to stick to the rule that v_i speaks on the i'th time window; rather, v_i can start its transmission window immediately upon completion of its listening window. Nevertheless, the solution is slightly complicated by the fact that failures can cause various links to transmit out of turn, making it difficult for a receiving node to identify with certainty the true starting time of its listening window.

In particular, each node v_i must listen all the time. On each round t, and for each of its incident links, v_i examines the messages it has heard on that link in the window of the last m rounds, $[t - m + 1, t]$. If $m/2$ identical copies of the same message have been received, then v_i accepts this message as a genuine one, and proceeds to start its own transmission window. By Chernoff's bound, the probability of receiving $m/2$ (or more) identical copies of a false message over some link during a window of m rounds, is exponenitally small.

Theorem 2.2 is complemented by the following negative result.

Theorem 2.3. *Assume malicious transmission failures that occur with probability $p \geq 1/2$. Then almost-safe broadcasting is not feasible in the message passing model even using a randomized algorithm.*

Proof. Consider a situation where $p \geq 1/2$. We prove the impossibility result via an adversarial argument, by presenting a graph, a probability distribution on the source messages and an adversary policy that will cause any

algorithm to err at least half the time. Specifically, consider a graph G consisting of two nodes, s and v, connected by an edge. The adversary will generate source messages for the sender s randomly from the set $\{0, 1\}$, fixing $M_s = 0$ or $M_s = 1$ with equal probability. We show the following:

$(*)$ The error probability of any broadcasting algorithm Π is at least 1/2.

Let us first note that it suffices to show $(*)$ for $p = 1/2$, since the adversary can always "slow" the failure rate at will. More precisely, we can reduce the case $p \geq 1/2$ to $p = 1/2$ as follows. If $p > 1/2$, then if the transmission is faulty in a given step, the adversary tosses a coin with heads probability $q = (p - 1/2)/p$ and delivers the correct message if heads turns up, otherwise exhibiting malicious behavior. Since $(1 - p) + pq = 1/2$, this is equivalent to a malicious adversary for $p = 1/2$.

We prove $(*)$ even assuming that only the unidirectional channel from s to v is failure-prone, and the channel from v to s is fully reliable. Without loss of generality we may assume that v simply echoes the messages it receives from s. Equivalently, the situation can be thought of as if communication is unidirectional, from s to v only, but s is able to listen on the channel, identify failures and know the precise value of the messages actually delivered to v in each step. This setting is equivalent to the original one because if v does send more elaborate replies to s using some protocol Π_v, then s can simulate those replies based on knowing the actual communication history.

We first prove $(*)$ in the simpler case of deterministic algorithms only and then extend the argument to probabilistic algorithms as well. Denote the sequence of messages delivered by s to v in the first k steps of the execution by σ. Following these k steps, and knowing σ, s now sends the next message, which is $A_0(\sigma)$ if $M_s = 0$ and $A_1(\sigma)$ if $M_s = 1$. If the transmission is faulty in the current step, then the adversary switches the sent message into the corresponding one for the opposite source message. More precisely, if $M_s = 0$, (hence s should send $A_0(\sigma)$), and a failure occurs, then the adversary delivers $A_1(\sigma)$ at v, and vice versa.

We now prove that the node v can never infer what is the original source message with probability better than 1/2. More formally, denoting by $\mathcal{M}_0$ (respectively, $\mathcal{M}_1$) the event that $M_s = 0$ (resp., $M_s = 1$), we show the following.

Claim 2.1. $\mathbb{P}(\mathcal{M}_0 \mid \sigma) = 1/2$ *for every execution* σ.

Proof. We prove this claim by induction on the number of steps. For zero steps the proof follows from the fact that $\mathbb{P}(\mathcal{M}_0) = 1/2$ due to the randomized policy used by the adversary for generating the source message. Now suppose the claim holds for every execution of k or fewer steps and consider an execution $\sigma' = \sigma \circ R$ of $k + 1$ steps. Let $\mathcal{R}_0$ (respectively, $\mathcal{R}_1$) denote the event that $R = A_0(\sigma)$ (resp., $R = A_1(\sigma)$). It is necessary to examine separately the cases $\mathcal{R}_0$ and $\mathcal{R}_1$ and prove the claim in either case. In fact, since the analysis is similar, we show only the case of $\mathcal{R}_0$. In this case,

$$\begin{aligned}\mathbb{P}(\mathcal{M}_0 \mid \sigma') &= \mathbb{P}(\mathcal{M}_0 \mid \sigma \circ R) = \mathbb{P}(\mathcal{M}_0 \mid \sigma \wedge \mathcal{R}_0) = \frac{\mathbb{P}(\mathcal{M}_0 \wedge \sigma \wedge \mathcal{R}_0)}{\mathbb{P}(\sigma \wedge \mathcal{R}_0)} \\ &= \frac{\mathbb{P}(\mathcal{R}_0 \mid \mathcal{M}_0 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_0 \wedge \sigma)}{\mathbb{P}(\mathcal{R}_0 \mid \sigma) \cdot \mathbb{P}(\sigma)}. \end{aligned} \tag{1}$$

We now claim that

$$\mathbb{P}(\mathcal{R}_0 \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_0 \mid \sigma). \tag{2}$$

To prove this, observe that if $A_0(\sigma) = A_1(\sigma)$ then both probabilities are 1, and if $A_0(\sigma) \neq A_1(\sigma)$ then both probabilities are $1/2$. In either case, Eq. (2) holds.

Combining Eqs. (1) and (2) implies

$$\mathbb{P}(\mathcal{M}_0 \mid \sigma') = \frac{\mathbb{P}(\mathcal{M}_0 \wedge \sigma)}{\mathbb{P}(\sigma)} = \mathbb{P}(\mathcal{M}_0 \mid \sigma) = \frac{1}{2} \tag{3}$$

by the inductive hypothesis, and we are done. ■

This proves $(*)$, and hence the algorithm is not almost-safe.

We now show how to extend our argument to arbitrary randomized algorithms. The adversary picks the same two-node graph G and the same uniform distribution of the source messages 0 and 1, as in the deterministic case. Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be the (possibly infinite) set of possible messages to be sent during any execution of the algorithm. Again denote by σ the sequence of messages sent by s to v in the first k steps of the execution. Following these k

steps, and knowing σ, the source s now chooses two probability distributions, $\mathcal{A}_0(\sigma) = \{(A_1, p_1), (A_2, p_2), \ldots\}$ and $\mathcal{A}_1(\sigma) = \{(A_1, q_1), (A_2, q_2), \ldots\}$, where p_i, q_i are probabilities, some of which can be equal to 0. If $M_s = 0$ then s randomly selects a message from the probability distribution $\mathcal{A}_0(\sigma)$ (i.e., it chooses message A_i with probability p_i, for $i = 1, 2, \ldots$) and transmits it. If $M_s = 1$ then s randomly selects and transmits a message from the probability distribution $\mathcal{A}_1(\sigma)$. This is what the algorithm prescribes and what happens when the transmission from s is fault-free in step $k+1$. If this transmission is faulty then the adversary switches the behavior of s into the corresponding one for the opposite source message. More precisely, if $M_s = 0$ and a failure occurs then the adversary delivers at v a random message using probability distribution $\mathcal{A}_1(\sigma)$ and if $M_s = 1$ and a failure occurs then the adversary delivers at v a random message using probability distribution $\mathcal{A}_0(\sigma)$.

As before, denoting by $\mathcal{M}_0$ (respectively, $\mathcal{M}_1$) the event that $M_s = 0$ (resp., $M_s = 1$), we show the following.

Claim 2.2. $\mathbb{P}(\mathcal{M}_0 \mid \sigma) = 1/2$ *for every execution* σ.

Proof. Again, the proof is by induction on the number of steps. For zero steps the proof is as before. Suppose the claim holds for every execution of k or fewer steps and consider an execution $\sigma' = \sigma \circ R$ of $k+1$ steps. Let $\mathcal{R}_i$ denote the event that $R = A_i$. Similarly to (1) in the deterministic case, we have the equation

$$\mathbb{P}(\mathcal{M}_0 \mid \sigma') = \frac{\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_0 \wedge \sigma)}{\mathbb{P}(\mathcal{R}_i \mid \sigma) \cdot \mathbb{P}(\sigma)}. \tag{4}$$

We also observe that

$$\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_1 \wedge \sigma) = \frac{p_i + q_i}{2},$$

hence

$$\mathbb{P}(\mathcal{R}_i \mid \sigma) = \frac{1}{2} \cdot \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) + \frac{1}{2} \cdot \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_1 \wedge \sigma) = \frac{p_i + q_i}{2},$$

and consequently

$$\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_i \mid \sigma). \tag{5}$$

Combining Eqs. (4) and (5), it follows that

$$\mathbb{P}(\mathcal{M}_0 \mid \sigma') = \frac{\mathbb{P}(\mathcal{M}_0 \wedge \sigma)}{\mathbb{P}(\sigma)} = \mathbb{P}(\mathcal{M}_0 \mid \sigma) = 1/2,$$

in view of the inductive hypothesis. ■

This completes the proof of the theorem. ■

Let us note that the proof of Theorem 2.3 relies strongly on the assumption that failures can cause the link to transmit out of turn. (Silence is considered as a message and a maliciously failed transmission can alter this message as well.) Without this assumption, the sender s can almost-safely broadcast a bit M to the receiver v so long as $p < 1$, using the following algorithm.

- Fix an integer $m > 1$.
- The sender s operates as follows:
  If $M = 0$, then transmit "hello" on steps $1, 2, 3, \ldots, 2m$.
  Else [$M = 1$], transmit "hello" on the even steps $2, 4, 6, \ldots, 2m$
  (and keep silent on the odd steps).
- The receiver v operates as follows:
  If received a transmission in any two consecutive rounds,
  then identify the message as 0.
  Otherwise, identify the message as 1.

It is easy to verify that if $M = 1$ then v always identifies it correctly. If $M = 0$ then v identifies it correctly in every execution containing at least two consecutive non-faulty rounds. This, in turn, is guaranteed whenever there were at least $m/2$ non-faulty rounds altogether, which by Chernoff's bound happens with probability at least $1 - \mathrm{e}^{-\Theta(m)}$.

2.2.3. Analysis for the radio model

We now turn attention to the radio model. We still use Algorithm Simple-Malicious and make the assumptions that each node v_i knows its index i and that all the nodes wake up simultaneously at round 0.

Theorem 2.4. *Assume malicious transmission failures that occur with probability $p < 1$. Then almost-safe broadcasting is feasible in the radio model if and only if $p < (1-p)^{\Delta+1}$, where Δ is the maximum degree of the network. Feasibility is established by a deterministic algorithm and impossibility holds for randomized algorithms as well.*

Proof. Consider a node v of degree $d \leq \Delta$, whose parent in the tree is w. Assume that w has a correct copy of the source message. Let $q = (1-p)^{d+1}$. By assumption $p < q$. Consider the $m = \lceil c \log n \rceil$ steps in which w is instructed by the algorithm to transmit. Let F_g be the event that v receives w's message correctly, and let F_b be the event that v receives w's message incorrectly. Let $F = F_g \cup F_b$ be the event that v receives w's message. We have $\text{Prob}(F_b) \leq p$ (as v can hear an incorrect message (namely, one different from the source message) from w only if the transmission from w fails) and $\text{Prob}(F_g) \geq q$ (since if all nodes from the neighborhood of v and v itself are fault-free in a given step, then v will hear the correct message). Hence $\text{Prob}(F_b|F) < 1/2$.

Consider the following events:

- E_{rec}: among the m steps there are at least $qm/2$ steps in which v receives w's message;
- E_{cor}: among the steps in which v receives w's message, the correct message is in the majority.

Let the constant c be such that the probability of $E_{\text{rec}} \cap E_{\text{cor}}$ is at least $1 - 1/n^2$. The existence of such a constant c follows easily from Chernoff's bound. Hence, with probability at least $1 - 1/n^2$, node v computes correctly the source message, provided that its parent did it correctly. It follows by induction that all nodes compute the source message correctly with probability at least $1 - 1/n$, which implies that Algorithm Simple-Malicious is almost-safe for the radio model, whenever $p < (1-p)^{\Delta+1}$.

For the other direction, consider a situation where $p \geq (1-p)^{\Delta+1}$. Consider a star graph G consisting of $\Delta + 1$ nodes, with the node v as the star root and the source s as one of the leaves. As for the message passing model, we prove the claim even under the assumption that s knows the precise value of the messages actually delivered to v in each step preceding the current one. Again, the adversary randomly generates source messages from the set $\{0, 1\}$ with equal probability.

Define the message sequence σ and the notations $\mathcal{M}_0$, $\mathcal{M}_1$ as before. As in the message-passing model, we prove that (for a suitably defined behavior of the adversary) the node v can never infer what is the original source message with probability better than 1/2, i.e., we show the following.

Claim 2.3. $\mathbb{P}(\mathcal{M}_0 \mid \sigma) = 1/2$ *for every execution* σ.

Proof. We prove this claim for any broadcasting algorithm, including a randomized one.

Consider such a broadcasting algorithm Π on G. Let σ be the sequence of messages heard by v in the first k steps of the execution of Π. Following these k steps, and knowing σ, the source s now chooses a (possibly infinite) set $\mathcal{A}(\sigma) = \{A_1(\sigma), A_2(\sigma), \ldots\}$ of possible messages to be sent in step $k+1$ (one of them can be the empty message, i.e., silence) and two probability distributions,

$$\mathcal{A}_0(\sigma) = \{(A_1(\sigma), p_1), (A_2(\sigma), p_2), \ldots\} \quad \text{and} \quad \mathcal{A}_1(\sigma) = \{(A_1(\sigma), q_1), (A_2(\sigma), q_2), \ldots\},$$

where p_i, q_i are probabilities, some of which can be equal to 0.

We prove the claim $\mathbb{P}(\mathcal{M}_0 \mid \sigma) = 1/2$ by induction on the number of steps. For zero steps the proof is implied by the fact that $\mathbb{P}(\mathcal{M}_0) = 1/2$. Now suppose the claim holds for every execution of k or fewer steps and consider an execution $\sigma' = \sigma \circ R$ of $k+1$ steps. We will describe a behavior of the adversary in step $k+1$ in which the only possible values of R are $A_i(\sigma)$, for $i = 1, \ldots, t$ and silence (i.e., the empty message), denoted by ϵ. Let $\mathcal{R}_i$, $\mathcal{R}_\epsilon$ denote the event that $R = A_i(\sigma)$ (resp., $R = \epsilon$). We examine separately the cases $\mathcal{R}_i$ and $\mathcal{R}_\epsilon$ and prove the claim in either case.

First suppose $R = A_i(\sigma)$. As for the message passing model, we have Eq. (4), hence it suffices to prove Eq. (5), as together with Eq. (4) we get $\mathbb{P}(\mathcal{M}_0 \mid \sigma') = 1/2$, thus completing the proof by induction.

Let S be the set of steps in which Π instructs s to transmit and all other neighbors of v, as well as v itself, to keep silent. First assume that the step $k+1$ is outside of S. In this case the adversary policy is the following: all faulty nodes behave as if they were fault-free, i.e., according to algorithm Π. Then the only possible message in this step is ϵ and hence $\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_i \mid \sigma)$ (this value being either 1 or 0, depending on whether $A_i(\sigma)$ is ϵ or not). Thus we may suppose that step $k+1$ is in S. Denote $q = (1-p)^{\Delta+1}$. By assumption we have $p \geq q$. As argued for the message passing model, the adversary may "slow" the failure rate as desired, hence in proving the claim we may assume $p = q$.

The adversary policy is the following:

- If s is faulty then all other faulty nodes keep silent and s switches its behavior to that corresponding to the opposite source message. More precisely, if $M_s = 0$ then s randomly transmits a message from the probability distribution $\mathcal{A}_1(\sigma)$, and if $M_s = 1$ then s randomly transmits a message from the distribution $\mathcal{A}_0(\sigma)$.
- If s is fault-free then all faulty nodes send a message different from the empty one.

We have

$$\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) \;=\; q_i\, p + p_i q.$$

Indeed, v can hear message $A_i(\sigma)$ either when s is faulty (and then $A_i(\sigma)$ will be heard with probability q_i) or when all nodes are fault-free (and then $A_i(\sigma)$ will be heard with probability p_i). Similarly,

$$\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_1 \wedge \sigma) = p_i\, p + q_i q.$$

Since $p = q$, we get

$$\mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_1 \wedge \sigma) = p(p_i + q_i).$$

Hence

$$\begin{aligned}\mathbb{P}(\mathcal{R}_i \mid \sigma) \;&= \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_0 \mid \sigma) + \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_1 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_1 \mid \sigma) \\ &= \frac{1}{2} \cdot 2p(p_i + q_i) = p(p_i + q_i) = \mathbb{P}(\mathcal{R}_i \mid \mathcal{M}_0 \wedge \sigma),\end{aligned}$$

which proves Eq. (5) for this case.

It remains to carry out the argument in the case $\mathcal{R}_\epsilon$. Now it suffices to prove $\mathbb{P}(\mathcal{R}_\epsilon \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_\epsilon \mid \sigma)$. Again, if step $k+1$ is outside of the set S, this is straightforward, provided that faulty nodes behave as fault-free. So assume that this step is in S and consider the adversary policy as in the case $\mathcal{R}_i$. If $A_j = \epsilon$ for some $1 \leq j \leq t$ then this case is covered by the previous argument. Hence assume that $A_j \neq \epsilon$ for all j. Consequently, regardless of whether the source message is 0 or 1, v hears ϵ precisely when s is fault-free and some other node is faulty. Let y be the probability of this event. Hence we have $\mathbb{P}(\mathcal{R}_\epsilon \mid \mathcal{M}_0 \wedge \sigma) = \mathbb{P}(\mathcal{R}_\epsilon \mid \mathcal{M}_1 \wedge \sigma) = y$, which implies

$$\mathbb{P}(\mathcal{R}_\epsilon \mid \sigma) = \mathbb{P}(\mathcal{R}_\epsilon \mid \mathcal{M}_0 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_0 \mid \sigma) + \mathbb{P}(\mathcal{R}_\epsilon \mid \mathcal{M}_1 \wedge \sigma) \cdot \mathbb{P}(\mathcal{M}_1 \mid \sigma) = \frac{y+y}{2}.$$

Thus we get Eq. (5) for this case too. ■

The theorem follows as before. ■

3. Complexity bounds

In this section we establish complexity bounds on the time of almost-safe broadcasting under the four considered scenarios. We consider only deterministic algorithms. As usual, the time of a given execution of a distributed deterministic algorithm in the synchronous model is defined as the number of time steps from the point when the first node started the execution until all nodes stopped their participation in the execution, and the time complexity of the algorithm (on instances of a given size) is the worst case time of any execution on such instances.

We first address the message passing model. Consider an n-node graph G with source s and let D denote the radius of the graph, i.e., the largest distance from the source to any other node. In the fault-free setting, the optimal broadcasting time is clearly D and is obtained by flooding. Hence D is a trivial lower bound on the time of almost-safe

broadcasting as well. Another lower bound is $\Omega(\log n)$. Indeed, if the source transmits at most $c \log n$ times, where $p^{c \log n} > 1/n$ then with probability larger than $1/n$ no node can receive the message, even for node-omission failures, and hence broadcasting cannot be almost-safe. On the other hand, we show an almost-safe broadcasting algorithm, for node-omission failures, working in time $O(D + \log n)$, and thus of optimal complexity.

The algorithm is based on the following result from [13].

Lemma 3.1 (*[13]*). *Consider a line of length L with a source at an endpoint. Suppose that each transmission may suffer a node-omission failure independently with probability $p < 1$. Then it is possible to broadcast in time $O(L)$ with probability at least $1 - e^{-cL}$, for any constant c.*

The algorithm from [13] is very simple. The authors present it for the 1-port model but its version for the message passing model (achieving the same performance) consists in all nodes transmitting simultaneously for $O(L)$ steps.

We modify this algorithm as follows. Let T be a breadth-first search tree rooted at the source (constructed by a centralized preprocessing algorithm). The height of T is D. Let $L = D + \lceil \log n \rceil$. Let all nodes of T transmit simultaneously for $O(L)$ steps.

For the sake of analysis, modify T by adding dummy nodes to each branch, so that all branches are of length exactly L. Denote the modified tree by T'. The execution of the algorithm on T can be compared with the corresponding execution on T'. By the analysis of [13], such execution has the property that for every branch, all of its nodes receive the message with probability at least $1 - e^{-cL}$, where c is such that $1 - e^{-cL} > 1 - 1/n^2$ (this is guaranteed by Lemma 3.1). As each branch in T is no longer than the corresponding branch in T', the probability that the message will reach all nodes on the actual branch in T is at least as large as the probability that the message will reach all nodes on the fictitious branch in T'. Hence all nodes of T get the message with probability at least $1 - 1/n$. This implies the following result.

Theorem 3.1. *For any $p < 1$, almost-safe broadcasting in any n-node graph of radius D with node-omission failures is possible in time $O(D + \log n)$. This complexity is optimal.*

For malicious transmission failures in the message passing model we get a slightly weaker result, applicable in a slightly weaker failure model where a failure cannot cause a link to speak out of turn (henceforth termed the *limited malicious* model). The algorithm is based on the following elegant result of Kučera [23], which concerns broadcasting a single bit in an unreliable environment.

Lemma 3.2 (*[23]*). *Consider a line of length L with a source at an endpoint. Suppose that each transmission may suffer a limited malicious transmission failure independently with probability $p < 1/2$. Then it is possible to broadcast in time $O(L)$ with probability at least $1 - e^{-\Omega(L^c)}$, for any constant $c < 1$.*

The algorithm of [23] uses single bit transmissions on the edges. It bounds also the *delay* of the algorithm, namely, the maximum time period during which a node is active in the algorithm (i.e., the time from when it gets the first bit from its predecessor in the line until it gets the last bit).

Denote the line of length n by $\mathcal{L}_n$. Let $A_p(n, \tau, \delta, Q)$ denote the fact that for $\mathcal{L}_n$, with individual transmission failure probability p, there exists a broadcast algorithm of time τ, with delay δ, guaranteeing failure probability at most Q. Clearly $A_p(1, 1, 1, p)$ holds. Suppose we are given an algorithm Π_n for the line $\mathcal{L}_n$ satisfying $A_p(n, \tau, \delta, Q)$. The algorithm is based on two basic composition rules, allowing the use of algorithm Π_n as a building block for constructing algorithms with improved guarantees.

The first composition rule applies to the line $\mathcal{L}_{\rho n}$ for integer $\rho \geq 2$, interpreted as a serial composition of ρ copies of $\mathcal{L}_n$. The algorithm for the composed line $\mathcal{L}_{\rho n}$ is based on using algorithm Π_n at times $j\tau$, for $j = 0, 1, \ldots, \rho - 1$, to broadcast from node jn to node $(j + 1)n$. It is shown that this composition operation guarantees

[CO1] If $A_p(n, \tau, \delta, Q)$, then $A_p(\rho n, \rho\tau, \delta, Q')$, where $Q' = 1 - (1 - Q)^\rho$.

The second composition rule is based on executing algorithm Π_n repeatedly κ times on the line $\mathcal{L}_n$, for integer $\kappa \geq 1$, with delay δ between successive executions. The last node of the line then takes the bit occurring in the majority of the messages it received as its output. It is shown that this composition operation guarantees

[CO2] If $A_p(n, \tau, \delta, Q)$, then $A_p(n, \tau + (\kappa - 1)\delta, \kappa\delta, Q')$, where $Q' = \sum_{j \geq \kappa/2}^{\kappa} \binom{\kappa}{j} Q^j (1 - Q)^{\kappa - j}$.

The algorithm is now constructed by carefully combining the two composition rules using suitable choices for the parameters ρ and κ.

In fact, there are three subtle differences between the setting of [23] and our setting. First, the message broadcast by that algorithm is comprised of a single bit, whereas we consider longer messages. Second, the failure model considered in [23] is slightly weaker than (limited) malicious. One may call that model the *flip* model, as the only transmission failures allowed are bit flips. In contrast, the arsenal of (limited) malicious transmission failures allows also for the possibility of message loss, as well as for the possibility of delivering the message intact and uncorrupted. This may in fact work to the advantage of the adversary in the case that the message bit has been flipped during some earlier transmission. The third difference is that the goal in the model of [23] is to deliver the correct bit to the *last* node on the line, whereas our definition of successful broadcast requires *every* node to receive the correct message.

Nevertheless, the algorithm of [23] is rather robust, and it can be readily verified that the claims **[CO1]** and **[CO2]** discussed above apply without change even when the broadcast algorithm is required to transmit long messages, overcome limited malicious transmission failures and guarantee correct message delivery at every intermediate node. Hence we may utilize this algorithm in our model as well.

To apply the algorithm of [23] on general networks, we modify it similarly to what was done earlier for node-omission failures. Find a breadth-first spanning tree T for the network centrally as before. Take any constant $\alpha > 1$, and let $L = D + d\log^{\alpha} n$ for a constant d to be determined below. All nodes of the tree T perform the algorithm from [23] on each branch. Whenever a node has more than one child in the tree, it transmits to all its children the message that it is instructed to transmit along the line in the original algorithm.

For the sake of analysis, consider a modified tree T' obtained by adding dummy nodes to each branch, so that all branches are of length L. By Lemma 3.2 (applied for $c = 1/\alpha$) the error probability in running the algorithm on T' is at most $e^{-bL^{1/\alpha}}$, for some constant b, and the time is at most $O(L)$. Let the constant d be such that $e^{-bL^{1/\alpha}} < 1 - 1/n^2$. Hence all nodes get the correct message with probability at least $1 - 1/n$. This implies the following result.

Theorem 3.2. *For any $p < 1/2$ and any constant $\alpha > 1$, almost-safe broadcasting in any n-node graph of radius D with limited malicious transmission failures is possible in time $O(D + \log^{\alpha} n)$.*

We now turn attention to the radio model. In this model, the natural benchmark for complexity of almost-safe broadcasting in a given n-node graph G with source s is the time opt of fault-free broadcasting in this graph. This time is clearly a lower bound on the almost-safe broadcasting time, as is $\Omega(\log n)$, for the same reasons as in the message passing model. Hence it is natural to ask whether almost-safe broadcasting is always possible in time $O(opt + \log n)$. Our next result reveals that this is not the case, even for node-omission failures, thus exhibiting a difference between the message passing and the radio models in this context (see Theorem 3.1). This is shown by constructing an n-node graph G for which $opt \in O(\log n)$ but any almost-safe broadcasting algorithm has time $\Omega(\log n \cdot \log\log n / \log\log\log n)$.

Consider a graph G constructed as follows. Let $N = 2^m$ for some integer $m \geq 1$. The graph G consists of three layers. The first layer, V_1, contains only the root, s. The second layer, V_2, consists of its neighbors, denoted $b_1, \ldots, b_m$. The third layer, V_3, consists of $N - 1$ nodes $1, \ldots, N - 1$. Altogether, the graph consists of $n = N + \log N$ nodes. The root is connected to every node in the second layer. Node b_i in the second layer is connected to every node v in the third layer such that the ith bit in the binary representation of v is 1.

Lemma 3.3. *In the radio network G, every algorithm for fault-free broadcasting from s must use at least $m + 1$ steps, and there is an algorithm achieving broadcasting in exactly $m + 1$ steps.*

Proof. Let us first prove that $m + 1$ steps are necessary. Without loss of generality, the source transmits in step 0 and then remains silent. We want to argue that the nodes of layer 2 require m steps to inform the nodes of layer 3. This is shown by induction on m. For $m = 1$ the claim is immediate. Now consider an arbitrary m. Consider the node $2^m - 1$. It hears the message only in steps in which exactly one of the nodes in layer 2 transmits. Hence there must be at least one such step. Without loss of generality let this be step 1, and let the transmitter be b_1. Then following this step, all the nodes $2^{m-1} \leq i \leq 2^m - 1$ have received the message, and we are left with the problem of informing the nodes $1 \leq i \leq 2^{m-1} - 1$. By the inductive hypothesis for $m - 1$, this will take $m - 1$ steps.

A schedule of length $m + 1$ realizing broadcast on G consists of the source transmitting in step 0, followed by m steps in which node b_i of layer 2 transmits in step i. ■

Lemma 3.4. *In the radio network G, any almost-safe broadcasting algorithm from s requires $\Omega(\log n \cdot \log\log n/\log\log\log n)$ steps, even with node-omission failures.*

Proof. Assume that the source is fault-free, so a single step of transmission by s suffices to inform all nodes of layer 2, and s remains silent thereafter. We want to argue that the nodes of layer 2 require $\Omega(\log n \cdot \log\log n/\log\log\log n)$ steps to inform all nodes of layer 3.

Consider an almost-safe broadcasting algorithm in the radio model with node-omission failures. Denote its schedule by $A_1, \ldots, A_\tau$, where $A_t \subseteq V_2$ is the set of indices of nodes from layer 2 that transmit in step t of the execution, for every $1 \leq t \leq \tau$.

For a node v of layer 3, let P_v denote the set of positions $1 \leq i \leq m$ in which the binary representation of v contains 1. Note that v can hear the transmission in step t iff $|A_t \cap P_v| = 1$. We denote this fact by defining a bit $H(v,t)$ such that $H(v,t) = 1$ if $|A_t \cap P_v| = 1$ and 0 otherwise. We say that v is *hit* in step t (or, by the set A_t) if $H(v,t) = 1$. Let h_v denote the number of steps in which node v is hit. Of course, v does not hear the message in such a step t if the unique node from layer 2 in $A_t \cap P_v$ fails in that step.

Claim 3.1. *With probability p^{h_v}, node v in layer 3 does not receive the message.*

This in turn implies the following.

Claim 3.2. *To guarantee that the broadcasting algorithm is almost-safe, it is required that $h_v \geq c \log n$ for every node v in layer 3, for some constant $c = c(p)$.*

Partition the nodes of layer 3 into sets S_j, such that $V_3 = \bigcup_{j=1}^{m} S_j$, where S_j consists of all the nodes $v \in V_3$ whose binary representation of v has exactly j ones. Let $h(t,j)$ denote the number of nodes of S_j that are hit in step t, i.e., $h(t,j) = \sum_{v \in S_j} H(v,t)$. Let $h(j)$ denote the number of hits on S_j throughout the execution, i.e., $h(j) = \sum_{t=1}^{\tau} h(t,j)$. By Claim 3.2, in order to ensure that the broadcasting algorithm is almost safe, it is required that for each set S_j, $h(j) \geq c \log n |S_j|$.

Let us reformulate the above in relative terms. Let $f(t,j)$ denote the fraction of nodes of S_j that are hit in step t, i.e., $f(t,j) = h(t,j)/|S_j|$, and let $f(j) = \sum_{t=1}^{\tau} f(t,j)$. In order to ensure that the broadcasting algorithm is almost safe it is required that for each set S_j, $f(j) \geq c \log n$.

By a straightforward combinatorial argument,

Claim 3.3. *A set A_t of cardinality ℓ satisfies $h(t,j) = \binom{\ell}{1} \cdot \binom{m-\ell}{j-1}$.*

Claim 3.4. *For a set A_t of cardinality ℓ, we have $f(t,j) \leq \frac{\ell \cdot j}{m}\left(1 - \frac{\ell - 1}{m-1}\right)^{j-1}$.*

Proof. By the previous fact,

$$f(t,j) = \frac{h(t,j)}{\binom{m}{j}} = \frac{\ell \cdot j(m-\ell)\cdots(m-\ell-j+2)}{m(m-1)\cdots(m-j+1)},$$

yielding the bound. ■

Let $K = \log m/\log\log m$ and $Z = \log K + \log\log K$. We now consider a set A_t of cardinality ℓ and a set S_j, and show that A_t can hit a sizable fraction of the nodes in S_j only if ℓ falls in the range $m/(jK) < \ell < m(Z+1)/j$. Specifically, outside this range A_t can hit at most an $O(1/K)$ fraction of the nodes of S_j.

Claim 3.5. *For a set A_t of cardinality ℓ:*

1. *If $\ell \geq m(Z+1)/j$ then $f(t,j) \leq 2/K$.*
2. *If $\ell \leq m/(jK)$ then $f(t,j) \leq 1/K$.*

Proof. The first claim follows from Fact 3.4 upon noting that if $\ell \geq m(Z+1)/j$ then

$$\ell \cdot j/m \geq (\ell - 1)(j-1)/(m-1) \geq \ell \cdot j/m - 1 \geq Z.$$

As the function $(x+1)e^{-x}$ is decreasing, we have that if $y \leq x' \leq x \leq x'+1$ then $xe^{-x'} \leq (y+1)e^{-y}$. Subsequently, we have

$$f(t,j) \leq \frac{\ell \cdot j}{m}\left(1-\frac{\ell-1}{m-1}\right)^{j-1} \leq (Z+1)e^{-Z} \leq 2/K.$$

For the second claim, note that if $\ell \leq m/(jK)$ then by Fact 3.4 again, $f(t,j) \leq \frac{\ell \cdot j}{m} \leq 1/K$. ■

Let $L(j)$ denote the set of integers ℓ such that a set A_t of cardinality ℓ satisfies $f(t,j) \geq 2/K$. By the previous fact, we have the following.

Claim 3.6. *If $\ell \in L(j)$ then $\frac{m}{jK} < \ell < \frac{mZ}{j}$.*

For $1 \leq i \leq K/4$, let $j_i = \lceil \frac{m}{(K(Z+1))^{2i-2}} \rceil$. Note that $j_1 = m$ and $j_{K/4} \geq 1$.

Claim 3.7. *For every step t, there is at most one index $1 \leq i \leq K/4$ for which $f(t,j_i) > 2/K$.*

Proof. The claim is proven by showing that the sets $L(j_i)$ for $1 \leq i \leq K$ are pairwise disjoint. To verify this, consider two consecutive indices j_i and j_{i+1}. By the previous fact, the largest ℓ in $L(j_i)$ is strictly smaller than $A_i = \frac{m(Z+1)}{j_i}$, and the smallest ℓ in $L(j_{i+1})$ is strictly larger than $A_{i+1} = \frac{m}{j_{i+1}K}$. Disjointness now follows from the fact that $A_{i+1} \geq A_i$. ■

The last fact implies that the contribution of each individual set A_t to the sum $F = \sum_{i=1}^{K/4} f(j_i)$ is less than $1 + (K/4-1) \cdot 2/K \leq 2$. As the total contribution of the entire schedule on these sets must be at least $cK \log n/4$, we get that

$$\tau > cK \log n/8 \in \Omega(\log n \log\log n/\log\log\log n),$$

completing the proof of Lemma 3.4. ■

Lemmas 3.3 and 3.4 imply the following result.

Theorem 3.3. *There exists an n-node graph for which almost-safe broadcasting in the radio model cannot be performed in time $O(opt + \log n)$, for both node-omission and malicious transmission failures.*

We finally present simple almost-safe broadcasting algorithms working in time $O(opt \cdot \log n)$, for any n-node graph whose optimal fault-free broadcasting time is opt. We first present the version for malicious transmission failures. Consider an optimal fault-free broadcasting algorithm A for a given graph. Let $p(v)$, for any node v, denote the node from which v gets the source message in algorithm A. Repeat every step i of A in a series S_i of consecutive $m = \lceil c \log n \rceil$ steps. Every node v which gets a message from $p(v)$ in step i of algorithm A, sets M_v to the majority of bits received from $p(v)$ in series S_i (or to 0, if the numbers of received zeroes and ones are equal). In later steps of the algorithm, when v is instructed to transmit, it transmits M_v. This scheme will be called Algorithm Malicious-Radio. In the version for node-omission failures, called Algorithm Omission-Radio, node v sets M_v to any bit received from $p(v)$ in series S_i (or to 0 if no bit was received).

Theorem 3.4. *Algorithm Omission-Radio is an almost-safe broadcasting algorithm working in time $O(opt \cdot \log n)$ for any n-node graph, for any $p < 1$. Algorithm Malicious-Radio is an almost-safe broadcasting algorithm working in time $O(opt \cdot \log n)$ for any n-node graph of maximum degree Δ, whenever $p < (1-p)^{\Delta+1}$.*

Proof. The proof of the first part is similar to the proof of Theorem 2.1 and the proof of the second part is similar to the proof of Theorem 2.4. ■

4. Open problems

While we establish exact feasibility constraints on almost-safe broadcasting for all investigated scenarios, the issue of the time complexity of almost-safe broadcasting is still far from being completely understood, and some interesting open problems concern tightening the bounds established in this paper. In particular we state the following problems.

1. Is there an almost-safe broadcasting algorithm for an arbitrary graph, working in time $O(D+\log n)$ in the message passing model with malicious transmission failures, when $p < 1/2$?
2. What is the optimal almost-safe broadcasting time for an n-node graph with optimal fault-free broadcasting time opt in the radio model with node-omission (resp. malicious transmission) failures? In particular, is it $\Theta(opt \cdot \log n)$?

Acknowledgements

The second author thanks Michael Brand for his helpful remarks. The first author was supported in part by NSERC discovery grant and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais. Part of this research was done during the second author's visit at the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

References

[1] Y. Afek, H. Attiya, A. Fekete, M. Fischer, N. Lynch, Y. Mansour, D. Wang, L. Zuck, Reliable communication over unreliable channels, J. ACM 41 (1994) 1267–1297.
[2] Y. Afek, B. Awerbuch, E. Gafni, Y. Mansour, A. Rosen, N. Shavit, Slide-the key to polynomial end-to-end communication, J. Algorithms 22 (1997) 158–186.
[3] N. Alon, A. Bar-Noy, N. Linial, D. Peleg, A lower bound for radio broadcast, J. Comput. Syst. Sci. 43 (1991) 290–298.
[4] B. Awerbuch, S. Even, Efficient and reliable broadcast is achievable in an eventually connected network, in: Proc. 3rd ACM Symp. on Principles of Distributed Computing, PODC, 1984, pp. 278–281.
[5] A. Bagchi, S.L. Hakimi, Information dissemination in distributed systems with faulty units, IEEE Trans. Comput. 43 (1994) 698–710.
[6] F. Bao, Y. Igarashi, K. Katano, Broadcasting in hypercubes with randomly distributed Byzantine faults, in: Proc. WDAG'95, in: LNCS, vol. 972, pp. 215–229.
[7] R. Bar-Yehuda, O. Goldreich, A. Itai, On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization, J. Comput. Syst. Sci. 45 (1992) 104–126.
[8] P. Berman, K. Diks, A. Pelc, Reliable broadcasting in logarithmic time with Byzantine link failures, J. Algorithms 22 (1997) 199–211.
[9] D. Blough, A. Pelc, Optimal communication in networks with randomly distributed Byzantine faults, Networks 23 (1993) 691–701.
[10] I. Cidon, I. Gopal, M.A. Kaplan, S. Kutten, A distributed control architecture of high-speed networks, IEEE Trans. Commun. 43 (1995) 1950–1960.
[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, Mc-Graw Hill Book Co., 1990.
[12] A. Czumaj, W. Rytter, Broadcasting algorithms in radio networks with unknown topology, in: Proc. 44th IEEE Symp. on Foundations of Computer Science, FOCS, 2003, pp. 492–501.
[13] K. Diks, A. Pelc, Almost safe gossiping in bounded degree networks, SIAM J. Discrete Math. 5 (1992) 338–344.
[14] D. Dolev, The Byzantine generals strike again, J. Algorithms 3 (1982) 14–30.
[15] M. Elkin, G. Kortsarz, Polylogarithmic additive inapproximability of the radio broadcast problem, in: Proceedings of 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX, in: LNCS, vol. 3122, 2004, pp. 105–116.
[16] A.D. Fekete, N. Lynch, Y. Mansour, J. Spinelli, The impossibility of implementing reliable communication in the face of crashes, J. ACM 40 (1993) 1087–1107.
[17] E. Gafni, Y. Afek, End-to-end communication in unreliable networks, in: Proc. 7th ACM Symp. on Principles of Distributed Computing, PODC, 1988, pp. 131–148.
[18] T. Hagerup, C. Rub, A guided tour to Chernoff bounds, Inform. Process. Lett. 33 (1989) 305–308.
[19] A. Herzberg, S. Kutten, Fast isolation of arbitrary forwarding faults, in: Proc. 8th ACM Symp. on Principles of Distributed Computing, PODC, 1989, pp. 339–353.
[20] C.-Y. Koo, Broadcast in radio networks tolerating Byzantine adversarial behavior, in: Proc. 23rd ACM Symp. on Principles of Distributed Computing, PODC, 2004.
[21] D. Kowalski, A. Pelc, Time of deterministic broadcasting in radio networks with local knowledge, SIAM J. Comput. 33 (2004) 870–891.
[22] E. Kranakis, D. Krizanc, A. Pelc, Fault-tolerant broadcasting in radio networks, J. Algorithms 39 (2001) 47–67.
[23] L. Kučera, Broadcasting through a noisy one-dimensional network, Tech. Report MPI-I-93-106, Max-Planck-Institut, Saarbruecken, Germany, 1993.
[24] O. Goldreich, A. Herzberg, Y. Mansour, Source to Destination Communication in the Presence of Faults, in: Proc 7th ACM Symp. on Principles of Distributed Computing, PODC, 1989, pp. 85–101.
[25] A. Pelc, Fault-tolerant broadcasting and gossiping in communication networks, Networks 28 (1996) 143–156.